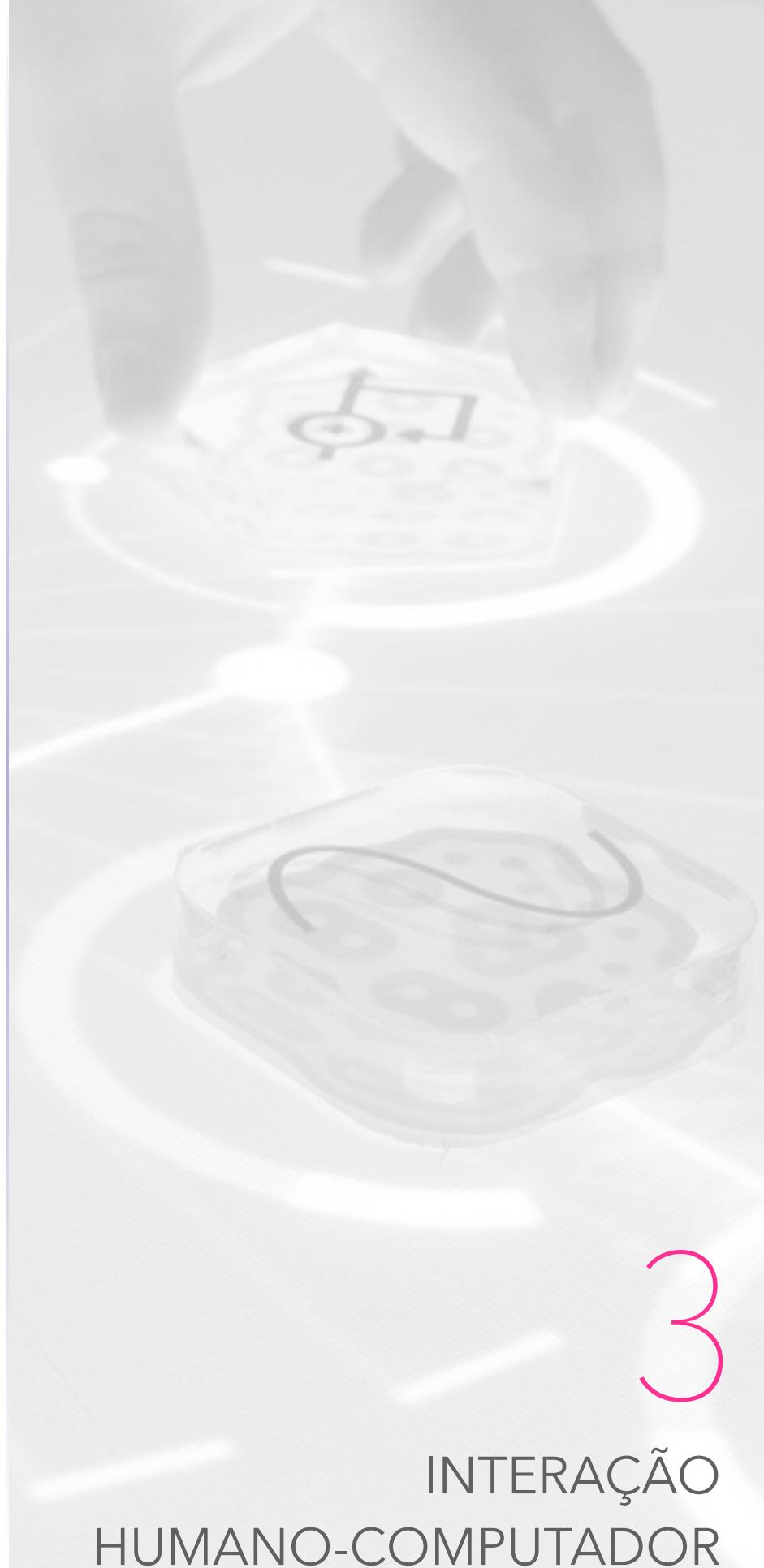


*If there's any object in human experience that's a precedent for what a computer should be like, it's a musical instrument: a device where you can explore a huge range of possibilities through an interface that connects your mind and your body, allowing you to be emotionally authentic and expressive.*

*Jaron Lanier*



universidade de aveiro  
teoria poiesis praxis



INTERAÇÃO  
HUMANO-COMPUTADOR

## **Atenção!**

---

Todo o conteúdo deste documento pode conter alguns erros de sintaxe, científicos, entre outros... **Não estudes apenas a partir desta fonte.** Este documento apenas serve de apoio à leitura de outros livros, tendo nele contido todo o programa da disciplina de Intereração Humano-Computador, tal como foi lecionada, no ano letivo de 2015/2016, na Universidade de Aveiro. Este documento foi realizado por Rui Lopes.

---

mais informações em [ruieduardofalopes.wix.com/apontamentos](http://ruieduardofalopes.wix.com/apontamentos)

Todos os dias as pessoas interagem com os mais variados sistemas, quer estes sejam eletrónicos ou não. A forma como ocorre a interação nem sempre é a mais natural - muitas vezes damos com a necessidade de nos confrontarmos com uma leitura prévia de um manual de instruções. Estas capacidades de tornar a forma de apresentação de um sistema aos humanos é a base do estudo desta área que durante anos que se tem expandindo, não mostrando prazo para limite - falamos claramente de interação humano-computador.

Consideremos um computador cuja **interface** é apenas um ecrã negro com dois botões: um que tenha a legenda "Sim" e outro com a legenda "Não". Sem termos a máquina ligada, como é que julgaríamos como é que o computador funciona? Talvez ao ligar apareça uma questão e ao utilizador apenas se pede que responda premindo os botões. Mas e se os botões tivessem antes as letras "S" e "N"? Julgaríamos a mesma coisa? Este raciocínio é equivalente às torneiras que usamos nas casas de banho. Todos nós temos por conceito que as cores azul e vermelho fazem respeito às temperaturas de frio e quente, respetivamente, mas olhando para a Figura 0.1, onde temos uma torneira de duas temperaturas da marca Kohler®, como é que abrimos a água quente ou a água fria?



**figura 0.1**  
interface de uma torneira

## 1. Princípios e Paradigmas de Usabilidade

Quando desenhamos um sistema temos de ter noção que nem tudo o que se desenha engloba o paradigma geral deste. Há um módulo que, qualquer que seja o sistema, nós, como programadores, não conseguimos gerir - falamos do **utilizador**. Um utilizador, neste paradigma, é denominado de **man-in-the-loop**, sendo um módulo muito complexo, não dominado pelo nosso conhecimento ou experiência e que não conseguimos controlar. Mais, há que ter a noção que para o utilizador mais comum, a interface é o próprio sistema - o utilizador não precisa de ter noção da complexidade algorítmica de um sistema, mas até pode atribuir uma complexidade ao funcionamento de uma interface.

Pelo facto de termos uma parte do sistema em que não temos conhecimento total da realidade de utilização, há que saber fazer aproximações corretas a este, através de **princípios de usabilidade** (que são independentes da tecnologia do momento, sendo que são mais relacionados com quem usa). Outro conhecimento a ter são os **paradigmas de usabilidade**, bastantes dependentes da tecnologia - a partir daqui podemos ter noção de exemplos de sucesso e com experiência *a posteriori* podemos ter também mais percepções acerca de métodos adequados para a criação de aplicações. Se juntarmos todas estas ideias chegamos a uma lista de **objetivos de usabilidade**. A **usabilidade** é assim uma característica que os sistemas podem ter, transversal a três dimensões: a **eficácia** (saber fazer ações sem erros), a **eficiência** (saber executar usando um número mínimo de recursos) e **satisfação**.

### Normas de utilização

Ao longo dos tempos tem havido cada vez mais a necessidade de saber interpretar e avaliar as condições com que vários materiais e equipamentos interagem com o ser

**utilizador**  
**man-in-the-loop**

**princípios de usabilidade**  
**paradigmas de usabilidade**  
**objetivos de usabilidade,**  
**usabilidade, eficácia**  
**eficiência, satisfação**

# 3

# INTERAÇÃO HUMANO-COMPUTADOR

humano. Quer sendo um sistema simples ou mais complexo, a forma como um determinado dispositivo se dispõe perante quem o utiliza deve ser igual, em grande parte das suas dimensões, de sistema em sistema. Para tal, no final da década de '80 começam a surgir inúmeras **normas de utilização** através, por exemplo, da **ISO** (*International Organization for Standardization*) para definir capacidades e requisitos que todos os equipamentos que interajam com humanos tenham de seguir. Dá-se o exemplo da norma ISO 9241-11, datada de 1998, que estabelece requisitos para equipamentos de escritório ergonómicos com terminais de dados visuais, onde um dos seus capítulos de desenvolvimento é responsável por indicar as melhores diretivas para uma orientação em termos de usabilidade, explicando como identificar quais as informações que um dispositivo nestas condições necessita de seguir para avaliar a usabilidade em termos de desempenho e satisfação.

Estas normas não são criadas uma vez e aplicadas para todo o sempre, sendo que a sua natureza permite que seja atualizada e revista a qualquer instante. Dá-se o exemplo da revisão da norma anteriormente vista para a ISO 13407 (referente a ISO 9241-210), datada de 2010, onde se centralizam os itens descritos anteriormente, num contexto mais próximo do humano para sistemas interativos. Esta norma também indica alguns princípios da própria área de desenho de sistemas que incorporam interação humano-computador, como uma presença muito ativa por parte de utilizadores, alocação apropriada entre funções de um sistema e os seus utilizadores, iteração de desenhos de solução e um desenho multi-disciplinar.

**normas de utilização, ISO**

## História em termos de paradigmas de usabilidade

Desde o início da história da computação que a usabilidade dos equipamentos se tem revelado como uma área fulcral para o desenvolvimento de qualquer sistema. Lembra-se, assim, que um sistema não serve para nada se a sua interface de interação com o humano não for legível, coerente ou coesa, de alguma forma.

Algures pelos anos '50 a interação com os computadores era feita através de interfaces dificilmente legíveis pelos humanos - falamos de **cartões perfurados** e, eventualmente, **fitas magnéticas**. Estes tipos de interface de nada serviam se não estivessem enquadrados num determinado contexto e a entrada de dados implicava que novos furos tivessem de ser feitos ou que uma gama de novos cartões tivesse de ser feita. Outra tecnologia da altura era o **teletype** (serviços de Telex) que usava papel como consola de interação com o humano, como podemos ver na Figura 1.1.

**cartões perfurados**  
**fitas magnéticas**

**teletype, Telex**



**figura 1.1**  
**teletype**

O tempo foi passando e na década seguinte e um novo conceito de interação surge: o **time-sharing**. Este time-sharing basicamente consiste na capacidade de uma só máquina possuir um conjunto de consolas que permitam a interação de vários utilizadores, executando diferentes funções em simultâneo

**time-sharing**

É só por meados da década de '70 que, pela investigação levada a cabo na Xerox PARC (especificamente Alan Kay), se cria e usa, pela primeira vez, uma interface gráfica via um ecrã, também denominado de **GUI** (sigla inglesa para *Graphical User Interface*), com o computador Xerox Alto. No entanto, da forma como hoje usamos, o primeiro dispositivo a incorporar um sistema com janelas, ícones, menus e ponteiros (vulgarmente designado de **WIMP**) foi o Apple Lisa e depois o Macintosh, ambos, que podemos ver na Figura 1.2.

**GUI****WIMP****figura 1.2**

Apple Lisa e Macintosh

Tendo a indústria se revolucionado sobre estes aspectos, nos anos '90, pela mão de Tim-Berners Lee, foi criada (ou aumentada) a rede que hoje todos conhecemos por Internet: a **WWW** (sigla inglesa para *World Wide Web*). A WWW é todo um universo de interação com o humano e que está em constante evolução.

**WWW**

## 2. Introdução às Interfaces de Utilizador

Todos os dias as pessoas interagem com sistemas considerados **interativos**. Tal designação provém da capacidade destes dispositivos responderem às ações que são conduzidas por meios de outrem. Neste capítulo iremos abordar alguns dos aspectos importantes a estudar, para quando estivermos integrados no papel da criação de uma interface de utilizador.

**interativos**

### Princípios de usabilidade

Como já tivemos oportunidade de referir, as interfaces que incorporam os sistemas, sejam eles operativos ou não, têm de seguir um conjunto de princípios que definimos como princípios de usabilidade. Aqui iremos falar de alguns, com exemplos em vários sistemas e em várias atualizações que se têm dado ao longo do tempo. Um primeiro princípio que podemos estudar é a capacidade que um sistema deve ter de indicar, nem que seja de uma forma aproximada, o estado atual do mesmo, fornecendo um **feedback**. Este feedback pode ser dado de forma mais incógnita, como o exemplo da variação dos ícones de reciclagem ou de bateria dos nossos computadores, ou através de formas mais visíveis, como uma barra de progresso ou legendas indicando mensagens acerca de estados do sistema. Comecemos por analisar o caso dos ícones de reciclagem. Na Figura 2.1 podemos ver a evolução dos ícones usados nos sistemas da Apple para indicar que a reciclagem está cheia ou vazia, usando a simbologia de um caixote do lixo cheio e vazio.

**feedback****figura 2.1**

feedback

## 5

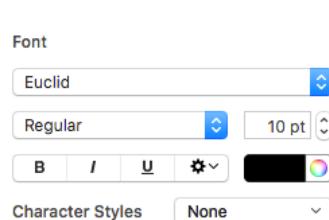
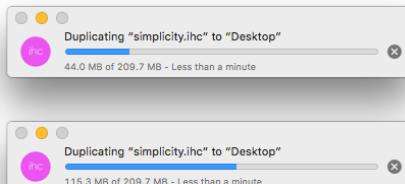
## INTERAÇÃO HUMANO-COMPUTADOR

Por outro lado, na Figura 2.2 podemos ver um exemplo de barra de progresso no sistema operativo Mac OS X, sobre o qual se aguarda a duplicação de um ficheiro.

Um outro princípio de usabilidade importante é o de **simplicidade**. A simplicidade é um fator importante para a boa e rápida utilização das ferramentas que um determinado sistema dispõe a um utilizador. Também é uma forma de dar a conhecer funções, ao utilizador, sem que este tenha que rever algum conceito ou refletir os seus conhecimentos num manual de instruções. Na Figura 2.3 temos o caso da interface de formatação de texto do processador de texto Pages, da Apple, em comparação com o painel de formatação nativo do sistema operativo Mac OS X, e a interface de formatação do processador de texto Microsoft Word para Mac OS X, em comparação com a sua janela de propriedades de formatação.

simplicidade

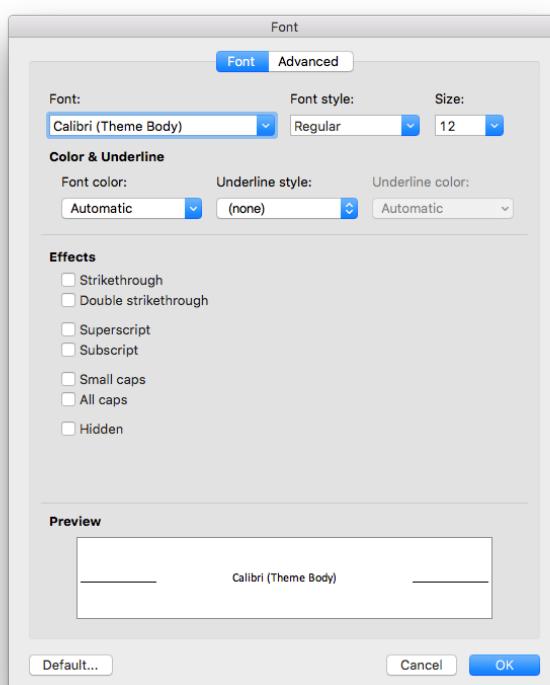
figura 2.2

painel de formatação  
(Pages 5.6.1)

Collection	Family	Typeface	Size
All Fonts	Estrangelo TurAbdin	Regular	10
English	Euclid	Italic	9
Favorites	Euclid Fraktur	Bold	10
Recently Used	Eurostile	Bold Italic	11
Diversão	Fabada		12
Largura fixa	Fira Mono for Powerline		13
Moderno	Fira Mono Medium for		14
PDF	Footlight MT Light		18

painel de formatação  
(nativo Mac OS X)

painel de formatação (Microsoft Word for Mac 15.11.2)



propriedades de formatação (Microsoft Word for Mac 15.11.2)

figura 2.3  
simplificação

Algumas das funções acessíveis nos painéis da Figura 2.3, seja em que sistema for, para muitos, já são conhecidas: quando vemos um botão com a letra ‘B’ já presumimos que se trate de “ativar negrito” ou quando vemos ‘I’ de “ativar itálico”. Quando referimos estas características dizemos que as aplicações necessitam de aspetos que lhe forneçam **familiaridade** com o utilizador, tirando o maior partido dos melhores comportamentos deste. Esta familiaridade, em grande parte, provém da representação de itens do mundo real numa realidade mais virtual - um caixote do lixo vazio e um cheio significa “reciclagem vazia” ou “reciclagem cheia”, respetivamente.

Em suma, podemos seguir alguns destes aspetos como sendo **heurísticas** para melhorar e *estandardizar* as nossas aplicações. Vejamos assim algumas heurísticas aplicáveis em projetos relacionados com interação humano-computador:

- visibilidade do estado do sistema - o sistema deve sempre manter o utilizador informado acerca do seu estado, através de mensagens apropriadas enviadas em tempo útil (*feedback*);
- proximidade entre o sistema e o mundo real - o sistema deve ser capaz de transmitir mensagens que coincidam com a linguagem do utilizador, com palavras, frases e conceitos familiares para o utilizador;
- controlo do utilizador e liberdade - o utilizador muitas vezes escolhe funções do sistema por engano e, para tratar o problema, deverá ser enviada uma mensagem devidamente orientada para a resolução até ao utilizador;
- *estards* e consistência - os utilizadores não deverão ter de saber que diferentes termos, situações ou ações significam o mesmo no contexto de uma determinada aplicação, pelo que todos os itens componentes de uma aplicação deverão seguir uma estratégia comum;
- prevenção de erros - mais do que boas mensagens de erro é saber evitar que o utilizador execute alguma função errada;
- reconhecimento de funções, mais que relembrar funções - tornar objetos e ações mais visíveis para incentivar o reconhecimento de funções pelo utilizador e não fazer com que este se tenha de lembrar da função;
- flexibilidade e eficiência de uso - usar aceleradores (impercetíveis para o utilizador mais recente/novato) que podem tornar a interação mais rápida com o hábito de utlizador (atalhos de teclado, ...);
- estética e *design* minimalista - as caixas de diálogo com o utilizador não devem conter informação que pode ser considerada irrelevante ou raramente necessitada;
- ajuda permite utilizadores reconhecer, diagnosticar e recuperar de erros - ao fazer mensagens de erro estas devem ser exprimidas em linguagem corrente (não orientada ao programador);
- ajuda e documentação - mesmo que a aplicação não precise, num primeiro contacto, de documentação, deve ser realizado um documento e a devida ajuda para uma determinada aplicação.

A avaliação das heurísticas é uma tarefa que acontece com a **experiência de utilização** (em inglês *user experience*, mais vulgarmente abreviado de **UX**). As boas ou más avaliações ocorrem neste processo porque o utilizador só interage com um determinado produto se tiver algum objetivo delineado para este. Este conhecimento, embora empírico, é geralmente mais valorizado que a própria usabilidade, dada a tamanha variação de estados de que é feito o utilizador. Assim, a experiência de usabilidade depende especialmente dos comportamentos dos utilizadores, tal como das suas atitudes ou emoções sobre a utilização de um determinado produto. Note-se que as conclusões a tirar de processos desta génese não são, de todo, determinísticos, sendo antes, na sua natureza, intrinsecamente dinâmicos.

**familiaridade**

**heurísticas**

**experiência de utilização,**  
**UX**

### 3. O Utilizador e Modelos Mentais

A primeira preocupação na criação de uma aplicação que interaja com um utilizador, seja esta um produto físico (*hardware*) ou virtual (*software*), deve estar concentrada na forma como a segunda parte (o **utilizador**) se sentirá aquando da sua utilização.

**utilizador**

#### Perfil de utilizador

O elemento mais complexo de todo o processo de desenho de um produto e testes é o utilizador. Mas porque é que referimos **complexidade**? Primeiro, de todas as partes constituintes de um projeto em computadores a única parte que não pode ser manipulada é o utilizador - ninguém pode, nem consegue, ter controlo sobre ele. Como unidade, este também diz-se complexo porque não há dois iguais - cada utilizador tem um gosto diferente, um comportamento diferente, atitudes diferentes e até mesmo emoções diferentes. Mas se tanta coisa é diferente, como saber designar o melhor em termos de interação humano-computador?

**complexidade**

Ao longo dos anos, foi-se tentando justificar e enquadrar todos os comportamentos, atitudes e emoções num só esquema prático - como se fosse um utilizador mediano. Tal enquadramento foi denominado, em inglês, de **HIPS** e é acrónimo de *Human Interface Processing System* (em português Sistema de Processamento de Interface Humana) e hoje serve como padrão para itens a ter atenção na realização de um projeto. Tais itens passam por estudar o possível **conhecimento** do utilizador (quer este seja lógico, dedutivo ou empírico), as **tarefas** do utilizador, algumas características **físicas**, o **ambiente** sob o qual o utilizador se encontra e algumas das **ferramentas** que este usa. Tudo isto permite que seja realizado um diálogo concordante (em grande parte) entre uma sistema aplicacional e um sistema utilizador.

**HIPS**

Estas conceções do HIPS, embora bastante subjetivas, têm base em estruturas reais e físicas do ser Humano. Afinal de contas, como estamos a falar dele, há que saber como é que este funciona. Podemos assim criar três grandes áreas de aproximação ao raciocínio e aceitação humana - uma área de estudo da componente percepcional, uma área de estudo de uma face mais cognitiva e, finalmente, uma área mais orientada para uma componente motora.

**conhecimento**

**tarefas, físicas**

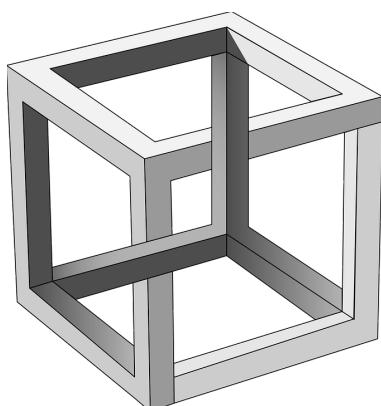
**ambiente, ferramentas**

Na **fase percepcional** é importante estudar o modo como os seres humanos captam e armazenam os **dados**. Em termos muito básicos, os mecanismos de leitura e armazenamento de dados é feita através de memória de elementos sensoriais (elementos visuais, auditivos, táteis, ...) e processamento de **reconhecimento de padrões**. Podendo ainda não ter percebido, os seres humanos, passam o tempo a reconhecer padrões em tudo o que os rodeia, tentando usar as suas capacidades preditivas para produzir ações. Este reconhecimento de padrões, por ser tão intuitivo, muitas vezes torna-se distorcido por outras fases de processamento de dados incorporadas na forma de pensar. Consideremos a Figura 3.1.

**fase percepcional**

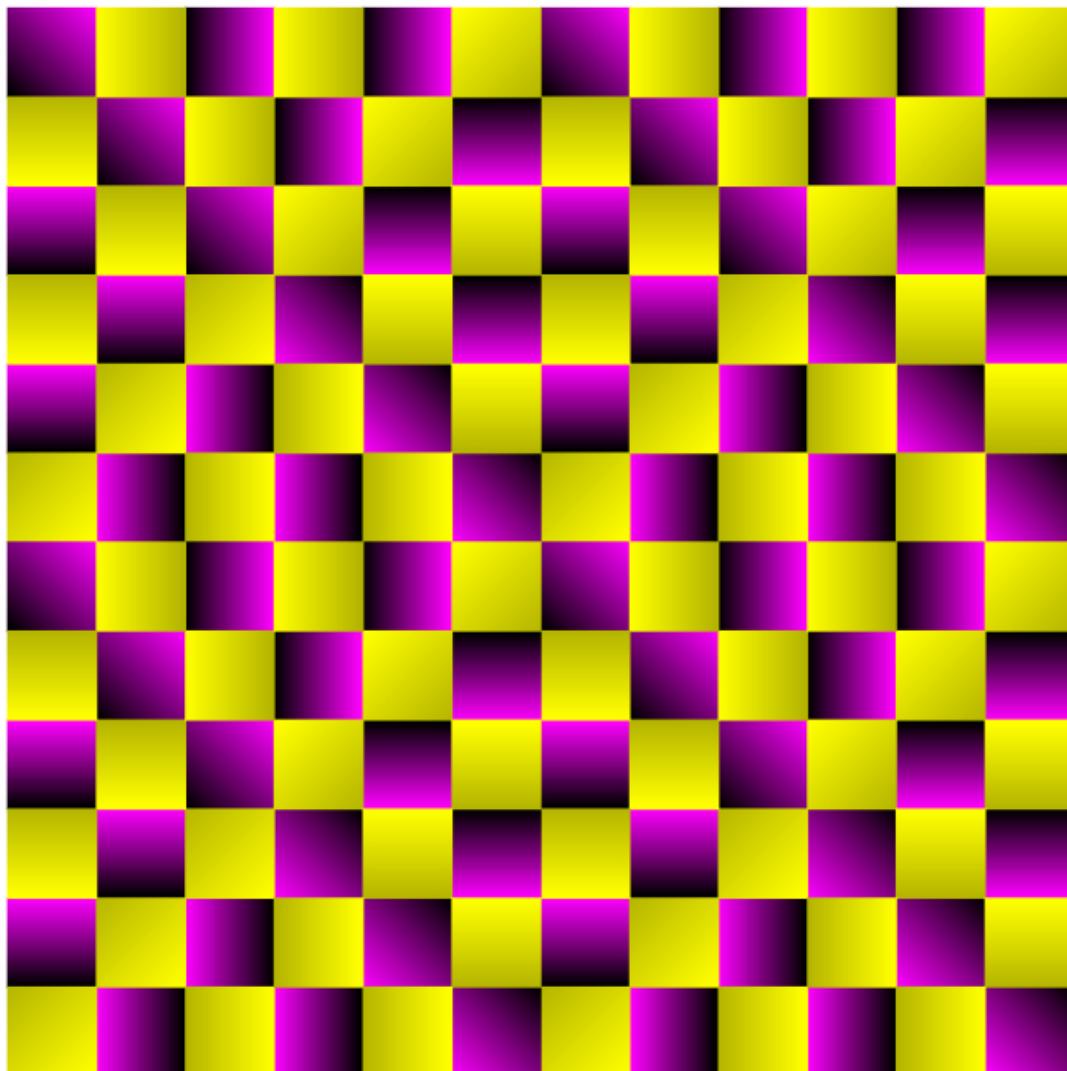
**dados**

**reconhecimento de padrões**



**figura 3.1**

Na Figura 3.1 podemos reconhecer um padrão logo à partida - um cubo. Numa segunda perspetiva reparamos que o cubo, embora nos continue a parecer um cubo, está numa posição estranha - não muito habitual. Com mais atenção podemos reparar que os vértices da diagonal espacial do cubo intersetam-se. A conclusão que obtivemos inicialmente, como podemos afirmar agora, foi totalmente precipitada e “inventada”. Vejamos um segundo caso, o da Figura 3.2.



**figura 3.2**

Na Figura 3.2, muito provavelmente, estar-se-á a ver uma imagem estática que parece estar-se a mover. Isto acontece porque a variação das gradientes que constituem a imagem provoca os nossos procedimentos de reconhecimento de padrões a preverem o comportamento de objetos tri-dimensionais (experimenta ver a imagem com um olho tapado e verás menos realce em termos de movimento - este é mais reduzido) e a ligarem as gradientes de elementos menores em elementos menores.

Como podemos ver pelas imagens em Figura 3.1 e Figura 3.2 o cérebro humano não é tão criterioso como “devia” em termos de processamento de dados (na leitura destes).

Numa **fase cognitiva** o ser Humano executa processamento de informação (recolhe dados e seletivamente processa-os em **informação**) e armazena os resultados. O armazenamento em memória, dependendo do nível de atenção e capacidade de retenção pode escolher ser efetuado sob uma secção denominada de **memória de curto-termo** (*short-term memory, STM*), quando recebe itens para armazenar como uma pequena sequência de números, de que necessite por poucos segundos, ou sob uma secção de **memória de**

**fase cognitiva**  
**informação**

**memória de curto-termo**  
**STM**

**longo-termo** (*long-term memory, LTM*), quando recebe um conjunto de itens com relações entre os vários objetos, como uma **rede semântica**. Estes conhecimentos podem ser usados no desenho de uma aplicação porque permitem focar a atenção de alguns componentes num utilizador. Consideremos, como exemplo, o caso da página Apontamentos, onde as disciplinas se encontram organizadas por cores, que designam áreas de estudo. Porque é que se optou por tal escolha? Veja-se a Figura 3.3.

**memória de longo-termo, LTM, rede semântica**

**figura 3.3**

<b>Introdução aos Sistemas Digitais</b>	isd
<p>A disciplina de Introdução aos Sistemas Digitais (a1s1) tem como programa curricular o estudo introdutório dos sistemas básicos de lógica aplicada a informática, tal como sistemas numéricos. Esta disciplina <b>coexiste</b> em engenharia de eletrónica e telecomunicações do DETI, sendo que se pode aplicar no estudo de Introdução à Arquitetura de Computadores de engenharia informática.</p> <p><b>ABRIR DISCIPLINA</b>    5 livros disponíveis    0 tutoriais disponíveis    0 exercícios disponíveis    0 soluções disponíveis</p>	
<b>Laboratórios de Informática</b> (a atualizar brevemente)	
<p>A disciplina de Laboratórios de Informática (a1) pretende dar aos alunos uma perspetiva global acerca de todas as tarefas ao encargo da informática e em estudo ao longo do curso. Esta disciplina <b>apenas existe</b> em engenharia de computadores e telemática do DETI.</p> <p><b>ABRIR DISCIPLINA</b>    1 livro disponível    0 tutoriais disponíveis    0 exercícios disponíveis    0 soluções disponíveis</p>	
<b>Laboratórios de Sistemas Digitais</b>	
<p>A disciplina de Laboratórios de Sistemas Digitais é bastante prática e leva os alunos a experimentarem modelar componentes lógicos numa placa de desenvolvimento baseada em FPGA. Esta disciplina <b>coexiste</b> em engenharia eletrónica e telecomunicações do DETI.</p> <p><b>ABRIR DISCIPLINA</b>    4 livros disponíveis    0 tutoriais disponíveis    0 exercícios disponíveis    0 soluções disponíveis</p>	
<b>Linguagens Formais e Autómatos</b>	
<p>A disciplina de Linguagens Formais e Autómatos (a2s2) tem como programa o estudo da geração e interpretação de compiladores e manipulação de linguagens formais. Semelhante à disciplina de Compiladores de Licenciatura em Engenharia Informática.</p> <p><b>ABRIR DISCIPLINA</b>    0 livros disponíveis    0 tutoriais disponíveis    0 exercícios disponíveis    0 soluções disponíveis</p>	
<b>Mecânica e Campo Eletromagnético</b>	
<p>A disciplina de Mecânica e Campo Eletromagnético (a2s1) tem como programa o estudo de vários sistemas físicos, mecânicos e elétricos e a sua relação com a área da computação. Semelhante às disciplinas de Mecânica Clássica e Campo Eletromagnético de MIEET.</p> <p><b>ABRIR DISCIPLINA</b>    1 livros disponíveis    0 tutoriais disponíveis    0 exercícios disponíveis    0 soluções disponíveis</p>	
<b>Matemática Discreta</b> (a atualizar brevemente)	
<p>A disciplina de Matemática Discreta é muito importante na área da informática para a caracterização dos paradigmas contextuais dos programas. Esta disciplina <b>coexiste</b> em Engenharia de Computadores e Telemática e Engenharia Informática.</p> <p><b>ABRIR DISCIPLINA</b>    0 livros disponíveis    0 tutoriais disponíveis    0 exercícios disponíveis    0 soluções disponíveis</p>	
<b>Métodos Probabilísticos para Engenharia Informática</b> (a atualizar brevemente)	
<p>A disciplina de Métodos Probabilísticos para Engenharia Informática (a2s1) tem como programa disciplinar a análise de problemas com recurso a métodos estocásticos de contagem e probabilidade. Esta disciplina <b>coexiste</b> na licenciatura em engenharia informática do DETI, tendo algumas semelhanças com a disciplina de Métodos Probabilísticos para Eletrotecnia de EET.</p> <p><b>ABRIR DISCIPLINA</b>    0 livros disponíveis    0 tutoriais disponíveis    0 exercícios disponíveis    0 soluções disponíveis</p>	

Ao escolher uma disciplina, primeiro o utilizador é confrontado com uma tarefa de pesquisa. Se este souber de antemão que a disciplina “Laboratórios de Sistemas Digitais” é da área de estudos de sistemas e possui a cor vermelha, então a atenção do utilizador é automaticamente focada no primeiro e terceiro item - sem querer, o utilizador exclui o segundo e os quarto a sétimo item. Do mesmo modo, para pesquisar “Matemática Discreta”, sabendo que pertence à área das matemáticas, de cor azul clara, o utilizador focar-se-á nos últimos dois itens. A este mecanismo damos o nome de **atenção seletiva**.

Em suma, o conceito HIPS é bom para descrever a forma como o ser Humano atua perante um objeto do qual precisa de utilizar - é uma forma de descrever as suas características emocionais, atitudes e comportamentais.

No entanto, há alguns pormenores que não podem ser descritos de forma tão global como num HIPS. Consideremos um universo de  $n$  utilizadores, com  $n >> 1000$ . Neste universo, quais são os efeitos do HIPS perante um sujeito com daltonismo, por exemplo. Tal indivíduo não conseguirá distinguir as cores do mesmo modo que outro conseguia e no HIPS tal descrição não poderá ser coerente com este caso. Note-se assim que o HIPS funciona para a grande generalidade dos casos, mas não serve para avaliar o universo de utilizadores como um conjunto de partes todas iguais.

**atenção seletiva**

## Influências do utilizador no desenho de interfaces

Com o HIPS (ou mesmo designando um único utilizador) que consequências deve ter a sua avaliação na interface a desenhar para um projeto de um produto? Consideremos vários casos: se um utilizador se revelar bastante conhecedor das tarefas de outros produtos da mesma classe que o projeto, então possivelmente esta aplicação não precisará de tanta ajuda e documentação para o assistir (não precisará de tanta documentação para aprender o que as funções do produto determinam, mas antes precisará de saber como executar determinadas ações) - diz-se assim que precisará mais de ajuda **sintática**; se um utilizador se revelar bastante conhecedor dos produtos da mesma classe do projeto, mas não tanto das tarefas que neles se desenvolve, então determina-se que há a necessidade de implementar mais ajuda sobre o que são as funções e não tanto sobre como encontrar as funções e saber julgá-las (deve haver, então, mecanismos de ajuda **semântica**).

**sintática**

Outras consequências, como se pode prever, são que se o utilizador usa uma determinada aplicação com bastante frequência, então a aplicação torna-se fácil de usar (e vice-versa).

**semântica**

## Modelos mentais de utilizador

O HIPS consegue descrever o comportamento, atitude e emoção de um utilizador mediano, mas como é que construímos um utilizador de teste? Como é que podemos criar critérios e objetivos a alcançar para agradar, mais tarde, os parâmetros descritos numa solução HIPS? Para tal criamos um novo conceito denominado de **modelo mental**. Um modelo mental é assim uma representação do sistema que é o utilizador e uma conceitualização e compreensão do mesmo, isto é, um resumo, em forma de modelo, do que é um utilizador, como age, o que espera e como sente - aqui não descrevemos uma entidade capaz de ter um comportamento, atitudes e emoções, mas antes cada uma destas.

**modelo mental**

Para criarmos um modelo mental, antes de mais, precisamos de testar o nosso produto. A criação é então gerada consoante as observações que são realizadas dos testes: conflitos, situações de agrado e desagrado, revoltas, ... Este tipo de conceito permite mais tarde efetuar previsões, modificando o próprio comportamento da aplicação em si e tornando-a mais flexível perante o mesmo ou novos utilizadores. Em termos de flexibilidade, pretende-se referir que um mesmo produto pode já ter a capacidade de prever gestos por parte do utilizador (ou de um utilizador em particular) e acionar outros mecanismos de prevenção de erros ou determinar outras causas de eventos que foram provocados. Esta passa a ser uma responsabilidade do gestor do projeto, ao analisar os vários utilizadores, que pode, mais tarde, passar para a aplicação.

No entanto, estes modelos mentais, embora possam parecer completos, para além de algo complexos de analisar, não são determinísticos (muito pouco rigorosos), instáveis e sem limites de julgamento, isto é, não tem noção criteriosa para avaliar se uma determinada causa que foi detetada é permitida ou não pelo sistema atual ou futuro - é impossível saber se de alguma forma o utilizador geral quererá que um determinado acontecimento volte a ser accionado mais tarde.

Para tornar estes modelos o mais lineares possíveis e assim conseguir tirar o máximo partido destes, há que adotar um *standard*, como as 10 heurísticas apontadas no capítulo anterior, permitindo que o utilizador consiga não só ser devidamente avaliado por um observador do projeto, como o próprio conseguir julgar-se a si próprio das suas capacidades e das da aplicação.

Como consequência do uso das heurísticas apontadas anteriormente, criam-se **metáforas**, isto é, pequenos modelos mentais implementados nos produtos que, num contexto virtual, aplicam formas e métodos de um contexto real. O caso do ícone da reciclagem é uma metáfora, dado que, embora não exista nenhum “caixote do lixo” no computador, a ideia de ser um “caixote do lixo” faz com que o ser Humano instantaneamente relate tal ícone com um local onde se pode eliminar o rastro de ficheiros e documentos. Note-se, no entanto, que uma metáfora pode ser um modelo mental que poderá aumentar o nível de complexidade de uma aplicação, se esta, de alguma forma, se encontrar incompleta. Por exemplo, nos sistemas operativos mais usados existe uma área vulgarmente denominada de “ambiente de trabalho” ou “secretaria”. Tal área pretende metaforizar o espaço de trabalho de uma pessoa no mundo real, mas em modo virtual, apresentando-se sempre como o primeiro ecrã ao ligar um computador e o pano de fundo para a execução de todas as janelas num ambiente gráfico. Não obstante, tal área caracteriza-se, em termos técnicos, como sendo um diretório (ou pasta) acessível de modo aberto e permanente. A ideia de poder abrir uma janela com o diretório “Secretária” (em inglês *desktop*) desconceitualiza a ideia de metáfora aplicada a este exemplo, tornando-o incompleto. Veja-se a Figura 3.4 onde se vê a “Secretária” no pano de fundo, como a mesa de trabalho no escritório, e a pasta “Secretária”, sob a forma de uma janela do explorador de ficheiros do sistema operativo em causa.

**metáforas**

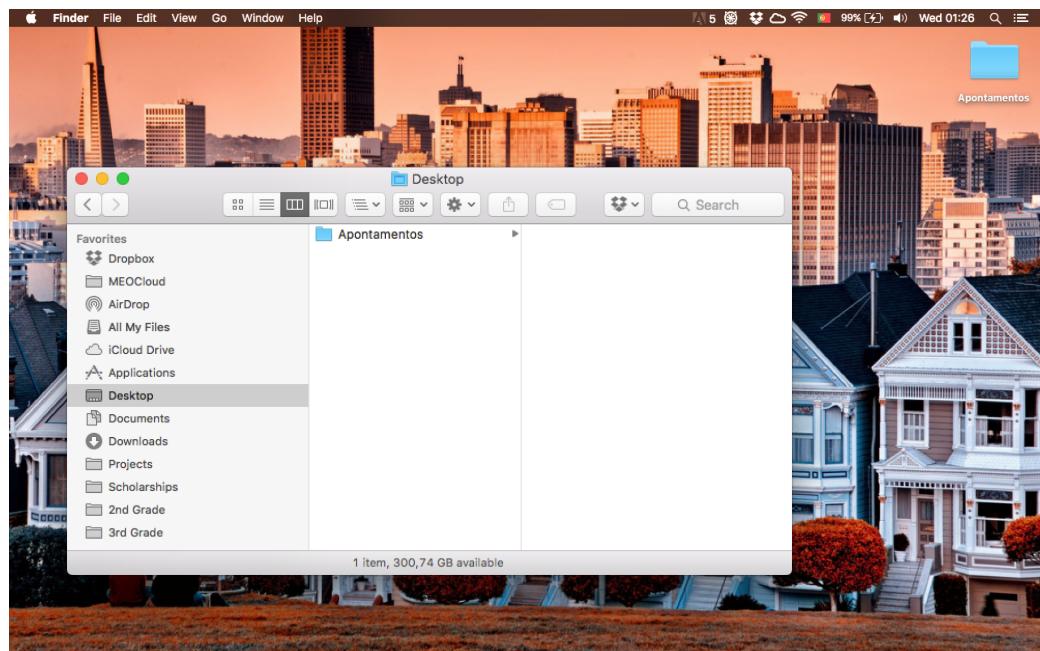


figura 3.4

Outros casos de limitações de metáforas podem ocorrer quando utilizadores menos experientes numa determinada aplicação precisam de metáforas mais explícitas do que aquelas que são fornecidas.

## 4. Diálogo entre computador e humano

Um produto, quer seja ele físico ou virtual, por muitas qualidades que possa vir a ter, não pode deixar de ter uma aspeto que é o mais importante de todos: capacidade de **comunicação** com o utilizador. Em termos de interação humano-computador nós designamos tal tarefa como **diálogo** e consiste em fazer transmitir uma mensagem ao utilizador sobre o estado atual, passado ou futuro da aplicação em uso.

**comunicação**  
**diálogo**

### Classes de diálogo

Em termos de aplicações computacionais os produtos devem ter uma larga variedade de diálogos, estes, que devem ser categorizados em várias **classes** (ou **estilos**). Podendo ser usadas em simultâneo, tais classes compreendem características em comum como os menus (listas de opções para desencadear ações), formulários (conjunto de campos de preenchimento para receber e levar a informação do humano até à aplicação (processamento central), teclas de função (atalhos de teclado únicos que desencadeiam ações diferentes por cada aplicação), linguagens naturais (interfaces de comunicação através da linguagem natural do utilizador, como inglês), entre outras.

**classes, estilos**

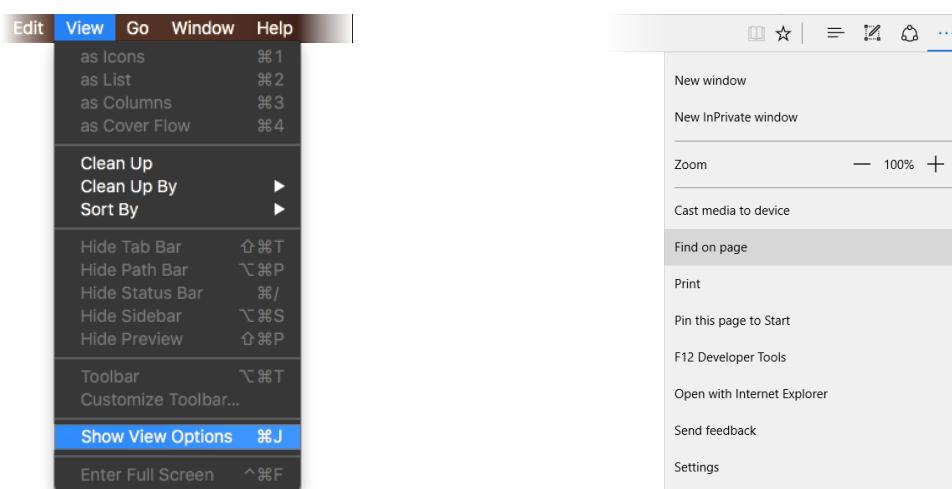
### Menus

Um dos estilos mais usados de diálogo entre as aplicações e os utilizadores são os **menus**. Análogos aos menus que nos são apresentados quando vamos a um restaurante, sobre os quais temos o painel de todos os pratos a escolher, por conseguinte, escolher um prato e fazer com que este seja cozinhado e trazido até nós, os menus das aplicações também se traduzem por uma vasta lista de opções de execução de funções. Existem vários tipos de menus, mas todos exibem um conjunto de escolhas, já pré-determinadas, para o utilizador decidir sobre o próximo estado de execução.

**menus**

Estas listas de opções podem ser apresentadas no ecrã sob várias formas. Uma mais conhecida é a **forma textual**, na qual se faz uma lista textual de ações para o utilizador escolher. Na Figura 4.1 podemos ver dois exemplos de menus textuais. Podemos, em tal figura, ver também que por vezes as opções textuais são acompanhadas de ícones, como é o caso de algumas aplicações para Microsoft Windows, contrariamente ao que acontece nas aplicações em Mac OS X.

**forma textual**



**figura 4.1**

Num paradigma de aplicações móveis estes menus também são possíveis, como podemos ver na Figura 4.2 proveniente de uma aplicação para o sistema operativo iOS, quer com ícones, como sem ícones.

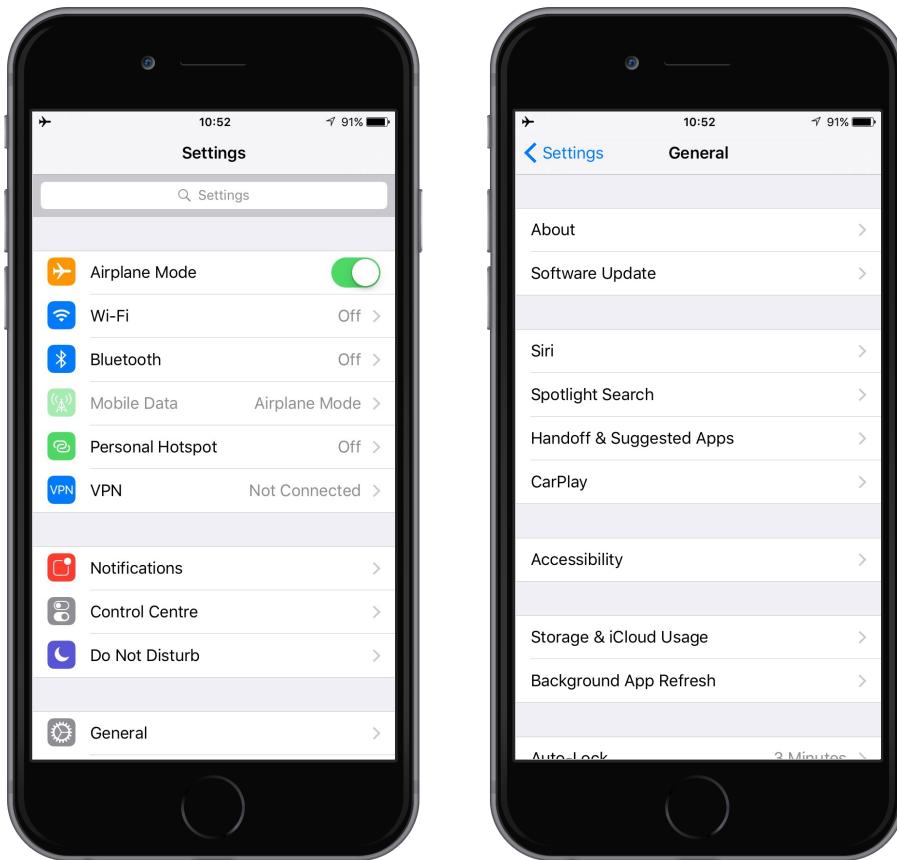


figura 4.2

Todos os menus apresentados nas imagens em Figura 4.1 e Figura 4.2 têm propriedades diferentes e algumas em comum: em comum, para além de serem menus, são todas (exceto a segunda imagem da Figura 4.1) propensas a ações em **cascata**, isto é, que têm opções que levam a novos menus - dá-se o exemplo direto da Figura 4.2, em que foi selecionada a opção “General”, que levou para a imagem da direita. Em termos de diferenças, podemos reparar que enquanto que o menu desta última figura não foi acionado por nenhum botão, que o fez apresentar-se no ecrã enquanto nenhuma opção é selecionada, no caso da Figura 4.1, ambas situações são de menus **pull-down/pull-up**, isto é, menus que se apresentam como uma lista de itens que aparece junto ao botão que o desencadeou. À posição onde o menu aparece relativamente a tal botão damos o nome de **âncora** e há situações em que este aparece por cima, como o Menu Iniciar (Figura 4.3), do Microsoft Windows 10.

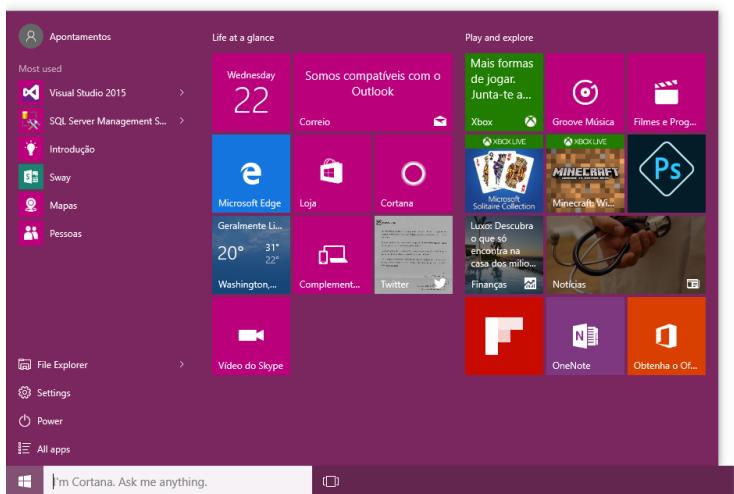
**cascata****pull-down/pull-up****âncora**

figura 4.3

Outros estilos de apresentação de menus encontram-se em situações **pop-up** como menus desencadeados por cliques em botão direito do rato, como podemos ver na Figura 4.4 (para o sistema operativo Mac OS X) ou como novas tecnologias como a 3D Touch<sup>1</sup>, onde o menu é desencadeado pelo toque com mais força numa determinada aplicação, como podemos ver na Figura 4.5.

figura 4.4

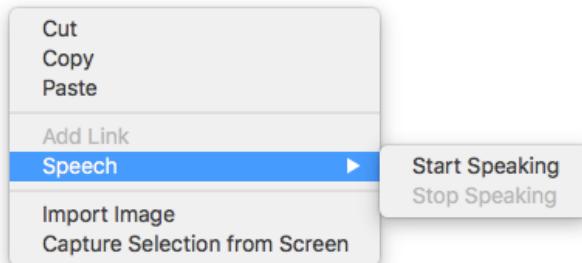


figura 4.5



Estes menus também podem estar presentes em janelas ou **caixas de diálogo**, onde se exibe uma janela na qual o utilizador poderá escolher 1 de  $n$  ações.

Como vantagens, tendo já uma ideia mais formal do que é um menu em termos de interface, podemos referir um grande potencial que é o facto de não exigir grandes explicações para quem utiliza, tomando partido da heurística referida anteriormente de reconhecimento ao invés de lembrar, o que facilita a execução de tarefas e evita erros ou ajudas sintáticas extra. No entanto, torna-se uma prática um pouco ineficiente e inflexível, dado que estamos a referir conteúdo que por si é estático (se mudar de estados alterna entre dois - ativo e inativo). Por esta mesma razão, empresas como a Apple têm dedicado tempo à investigação de novas abordagens de menus, como conteúdo dinâmico apresentado

caixas de diálogo

<sup>1</sup> 3D Touch é uma tecnologia registada da Apple Inc..

nos menus desencadeados por 3D Touch ou na alteração de especificidade das opções de um menu, em Mac OS X, premindo apenas a tecla “option” (⌥), como podemos ver na Figura 4.6.

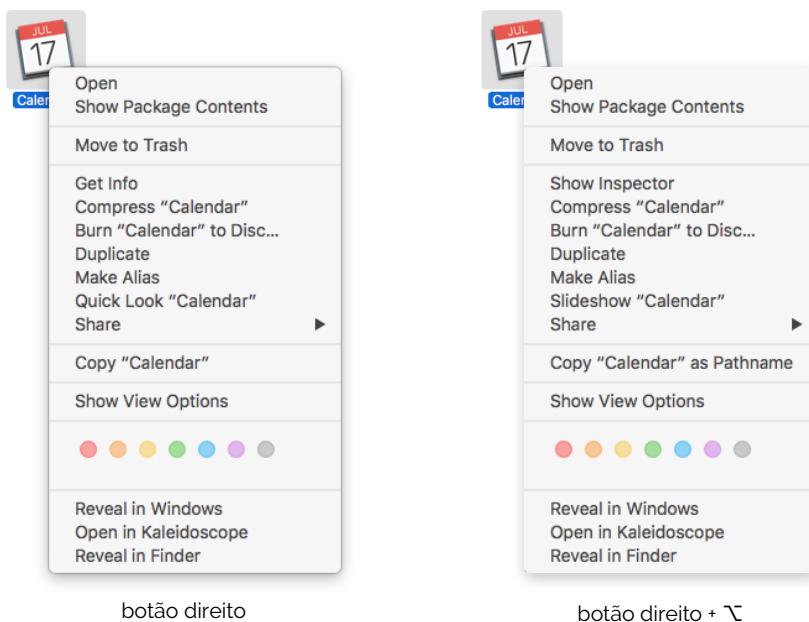


figura 4.6

Os menus são claramente ótimos para utilizadores cuja experiência de uso da aplicação é baixa (tanto experiência de sistema como de tarefas), utilizadores que usam diferentes tipos de sistemas ou que sustêm poucos conhecimentos no domínio da computação.

Na criação de um menu é importante saber designar uma **ordem** pela qual os itens aparecem no ecrã, e até mesmo saber dividir várias áreas de abordagem, como podemos ver na Figura 4.6, onde a atribuição de etiquetas está num sítio diferente do abrir a aplicação ou mostrar os seus conteúdos, por exemplo. Em suma, e mais concretamente, podemos seguir o esquema da Figura 4.7 para selecionar uma ordem mais adequada.

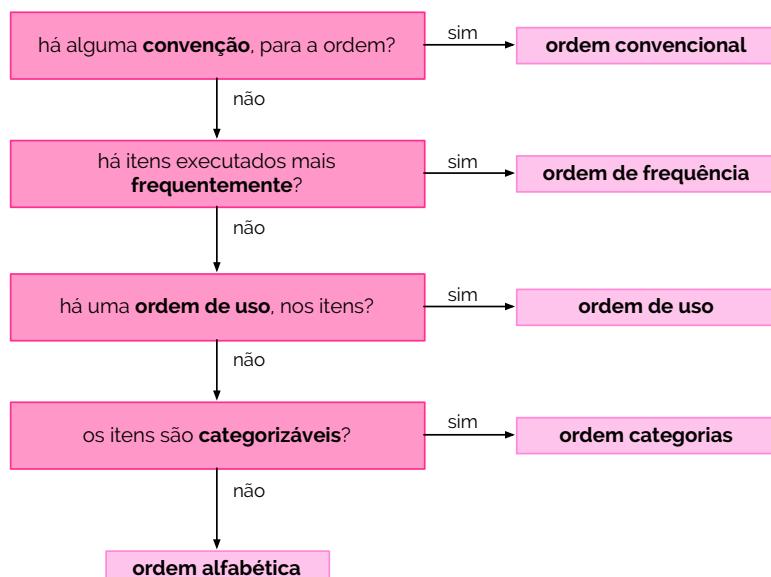
**ordem**

figura 4.7

ordem em menus

Alguns menus incluem ainda outro tipo de opções de entrada: **escolhas marcadas**. Entre os dois tipos de escolhas marcadas temos: **radio buttons** (botões radiais), que entre  $n$  botões apenas permite a escolha de 1; e **checklist items** (itens de verificação), que

**escolhas marcadas**  
**radio buttons**  
**checklist items**

denotados com um visto, permite que de entre  $n$  itens sejam escolhidos  $n$ . Na Figura 4.8 podemos ver ambos exemplos e na Figura 4.9 uma aplicação dos mesmos.

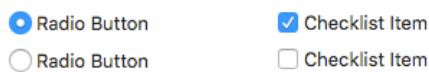


figura 4.8

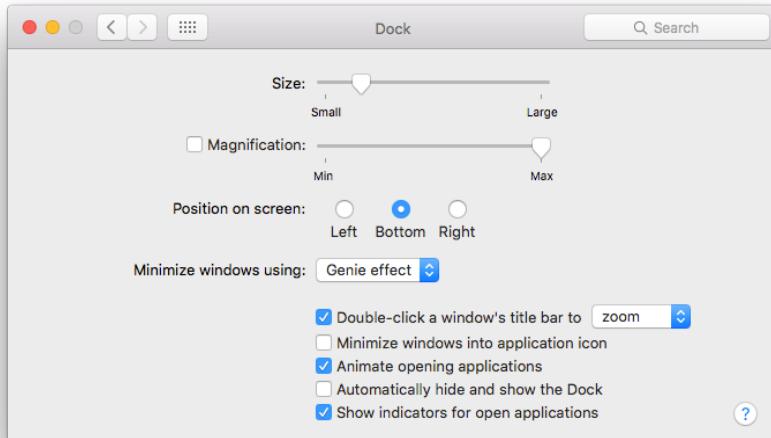


figura 4.9

Uma outra possibilidade de melhorar os menus é adicionando-lhes **tooltips** (ajudas breves) ao sobrepor o cursor no item do menu. Esta ajuda pode ser o nome de uma função (ao sobrepor o cursor num ícone) ou uma descrição breve de uma ação (ao sobrepor a uma função mais composta). Veja-se a Figura 4.10.

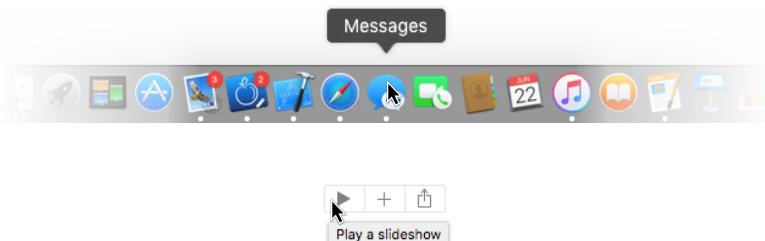
**tooltips**

figura 4.10

## Linguagens naturais

Uma segunda forma de diálogo, tal como já foi referido, é através da **linguagem natural**, isto é, tentar fazer com que tanto o computador, como o humano, consigam interagir um com outro através de um idioma, como o inglês. Sistemas como a Siri (da Apple), Google Now (da Google) ou Cortana (da Microsoft) são três dos sistemas possíveis de permitir a interação com os utilizadores através da voz, interpretando linguagem inglesa, por exemplo. Veja-se a Figura 4.11 onde se exibe a Cortana em funcionamento (em Windows 10) e a Figura 4.12 onde se exibe a Siri em funcionamento (em iOS).

**linguagem natural**

Este tipo de interação, no entanto, tem-se mostrado cansativo, dado que a velocidade de processamento das mensagens ainda não é suficientemente rápida para que haja uma interação fluida - e no sentido humano percebe computador muitas vezes a leitura não é feita de modo correto dada a não-natureza da linguagem por parte do computador.

Outros sistemas que incluem esta capacidade de diálogo por linguagem natural é o caso do Mac OS X que possibilita, como ferramenta de acessibilidade, a exibição de erros narrados por uma voz do sistema (Figura 4.12) ou a ferramenta de ditado (*speech-to-text*) (Figura 4.13).

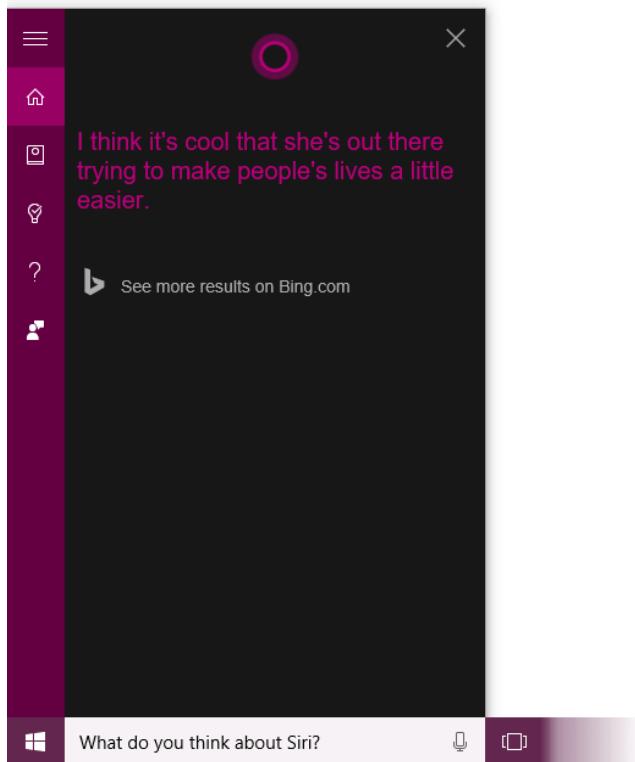


figura 4.11



figura 4.12



figura 4.13

This is me talking with the computer |

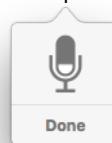


figura 4.14

Em suma, para criar um sistema que possua um serviço de diálogo por linguagem natural há que ter em consideração que é necessário garantir, de qualquer forma, que a linguagem natural coincida com o público-alvo do produto em projeto. Se um determinado telemóvel, por exemplo, possuir um sistema desta classe, há que ter o cuidado de implementar técnicas que permitam que o produto suporte idiomas variados, dado que, decerto, o telemóvel não será utilizado exclusivamente por um universo de falantes de um só idioma. Uma possível solução para este problema pode passar por **restringir** o fecho da linguagem natural, isto é, criar um sub-conjunto da linguagem de partida (um idioma) e aplicá-lo ao contexto da aplicação.

restringir

Este tipo de aproximação de diálogo é ótimo para pessoas que necessitem de detalhes de acessibilidade como suporte à utilização de uma máquina, assim sendo, pessoas com alguma experiência para usar tais ferramentas. No entanto, com pouco treino consegue-se obter uma boa relação com esta técnica.

## Linguagem de comandos

Numa perspetiva mais próxima do prisma de um programador, uma alternativa de diálogo humano-computador é a **linguagem de comando**, isto é, a transmissão de mensagens por via de linguagens reduzidas e sumárias, baseadas em átomos representados por palavras simbólicas para ações.

linguagem de comando

Tal como já devemos ter experiência, linguagens deste tipo são úteis para executar rapidamente processos, mas têm um forte requisito para quem as usa: os utilizadores precisam de portar uma experiência algo avançada para entrar nestes sistemas. Alguns aspectos importantes a estudar numa abordagem deste tipo são a sintaxe e semântica das mensagens.

Nos sistemas UNIX já devemos ter conhecimento de alguns comandos que nos permitem fazer as ações mais simples, como `rm` para remover um ficheiro ou `rmdir` para remover um diretório. Como podemos ver, e temos muitos outros exemplos ao longo de

toda a linguagem que compõe as consolas UNIX, os comandos são extremamente minimalistas e autoesclarecedores. No entanto, para acedermos a um painel de ajuda também é necessário que o utilizador seja capaz de atender apenas o que precisa (saiba filtrar a informação toda), dado que nestes sistemas as informações são bastante detalhadas e são dadas todas em simultâneo.

## Formulários

Sempre que num serviço há a necessidade, por parte da aplicação, de pedir muita informação a um utilizador, uma excelente forma de o fazer é criando um diálogo através de um **formulário**. Outrora sendo a única forma de diálogo, no início da interação humano-computador, este sistema acaba por ser autoexplicativo e facilita o utilizador na inserção de dados, informando-o, sempre que necessário, acerca da melhor forma de preencher os campos, como datas, moedas, entre outros. Veja-se a Figura 4.15 onde se exibe um formulário.

**formulário**

Email \*

Full name \*

Your legal name, used for any certificates you earn.

Public username \*

The name that will identify you in your courses - (cannot be changed later)

Password \*

Country \*

Gender

Year of birth

Highest level of education completed

Tell us why you're interested in edX

I agree to the edX [Terms of Service and Honor Code](#). \*

**Create your account**

**figura 4.15**

Na Figura 4.15 temos o formulário de inscrição do serviço web edX, no qual podemos verificar que até o utilizador menos experiente consegue preencher devidamente os campos. No entanto, também podemos verificar que este tipo de diálogo pode permitir que erros se propaguem, vindos do utilizador, e, fazendo uma razão entre o tamanho desta página e o tamanho da imagem, ocupa muito espaço no ecrã.

Note-se que para que se crie um sistema com um formulário há que saber gerar uma boa forma de apresentar os campos a preencher, determinando uma **ordem**. Esta ordem deve determinar do primeiro campo para o último o grau de especificidade semânticos, isto é, deverá ser do campo mais particular (específico) ao mais global ou vice-versa.

**ordem**

## Teclas de função

Embora mais orientado para utilizadores mais experientes com determinados sistemas de operação, as **teclas de função** são outra forma de diálogo humano-computador. Nesta forma a comunicação entre o humano e o computador (só se faz nesta direção, por enquanto) é mais rápida.

**teclas de função**

Existem dois tipos de teclas de função - as *soft keys* e as *hard keys*. Quando referimos **soft keys** pretendemos as teclas de função que, no teclado, apresentam as letras  $F_x$ , sendo  $x$  um dos valores do seguinte conjunto  $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$ . Estas teclas dependem da aplicação que está aberta e (geralmente) podem ser personalizadas (de forma nativa aos softwares). As **hard keys** são as “ctrl”, “alt”, “option”, “cmd”, logótipo do Windows, “shift”, ... cujas funções não se alteram de aplicação em aplicação, tendo uma função ligada ao sistema de operação.

**soft keys****hard keys**

## Manipulação direta

Um dos métodos de diálogo mais intuitivos para o ser Humano é a **manipulação direta**. Neste método o utilizador tem a capacidade de interagir de forma mais próxima com o sistema aplicacional, na forma de representações visuais. Temos assim uma representação contínua de objetos, onde substituímos linguagens de comando por ações físicas, podendo assim, por cada gesto, ter um *feedback* instantâneo e rápido.

**manipulação direta**

Exemplos deste tipo de métodos são as seleção de um texto num processador de texto, o destaque de itens do ecrã quando um cursor os sobrepõe, aplicações de desenho gráfico, aplicações de **realidade aumentada e virtual** (aplicações onde coexiste uma representação de um sub-mundo virtual).

**realidade aumentada e virtual**

Com esta técnica de diálogo surgiram novos paradigmas que até então não estavam incluídos no estudo de interação humano-computador: um dos quais, e possivelmente o mais importante, é a distância articular e a semântica. A **distância articular** é a distância entre o significado das ações e a sua forma física. Já a **distância semântica** é a distância entre o objetivo do utilizador em concretizar uma ação e a semântica da interface. Em suma, e tendo em vista uma aplicação, num programa de processamento de texto a distância semântica para efetuar um negrito é bastante reduzida (o utilizador consegue ver um ícone que lhe permite colocar uma palavra a negrito, muito facilmente). Em termos de distância articular consideremos o caso do arrastar uma aplicação para ser instalada no sistema Mac OS X. A distância que o utilizador prevê para a instalação é a articular, obtida desde o momento em que o utilizador olha para a interface e interage com ela para mover a aplicação.

**distância articular**  
**distância semântica**

Este método, assim, pode ser visto como fácil de aprender a interagir e lembrar, usando uma filosofia de WYSIWYG (*What You See Is What You Get*, em português, “o que vês é o que obténs”).

## 5. Apresentação em Ecrã

O principal meio de comunicação do computador para o humano é o **ecrã** - é para ele que o utilizador olha para ver o estado atual do sistema computacional. Assim sendo é importantíssimo saber como **dispor** todos os itens no ecrã e de que forma os por.

**ecrã****dispor**

## Estrutura (Layout)

Num nível de abstração mais elevado devemos ser capazes de visualizar, em todo o projeto visual no ecrã, a noção de **estrutura** (em inglês vulgarmente denominado de **layout**). Por outras palavras, devemos ser capazes de distinguir porções do projeto, de uma forma regular, ao longo de todo o trabalho.

Tais porções de projeto devem ser separadas entre elas e devidamente categorizadas segundo uma determinada lógica. Na nossa cultura ocidental, devendo tomar a origem do plano no ponto mais elevado e à esquerda, os blocos de informação deverão conter toda a informação que é estritamente necessária a partir desse ponto. Veja-se a Figura 5.1 onde temos uma página web de um jornal americano, o USA Today.

**estrutura**  
**layout**

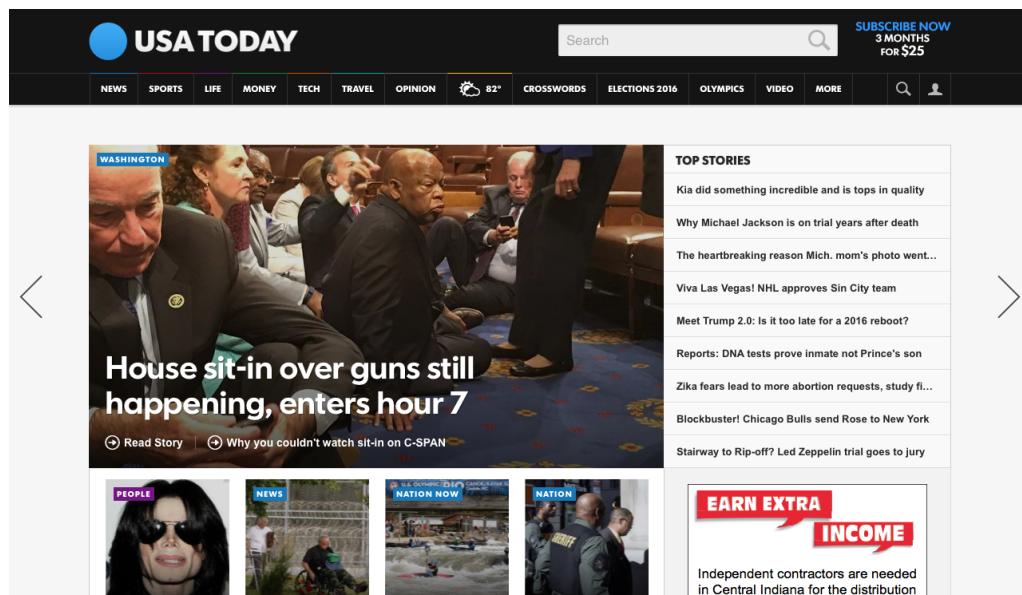


figura 5.1

A forma como os itens são apresentados no ecrã, na Figura 5.1, nota-se que é baseada numa estrutura rígida. Na Figura 5.2 podemos ver melhor as várias divisões (porções) que constituem a página.

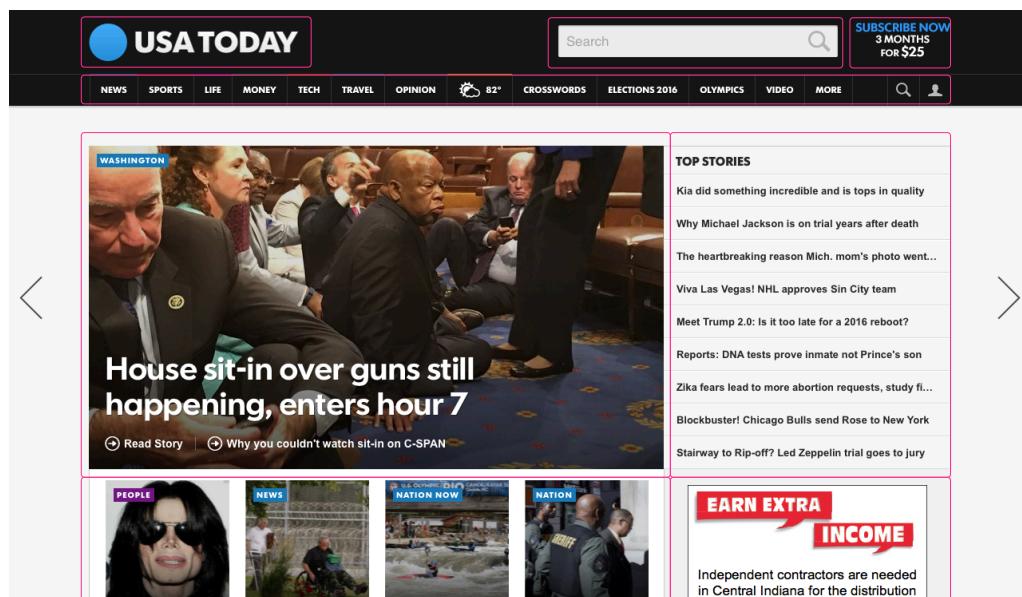


figura 5.2

Como podemos ver temos grupos que servem para cobrir, num primeiro nível, a abstração da estrutura-base da página. Vemos então um grupo para o logótipo, outro para a caixa de pesquisa, para a barra de menu, para o destaque, ... Se reparmos bem, os itens que estão diretamente relacionados mantém um determinado estilo (vide os vários destaques da página à direita “top stories”, por exemplo). Embora localmente haja um termo que possa estar a negrito ou outros que possam ter um pouco mais de letras maiúsculas, por norma, para que uma estrutura (*layout*) seja agradável ao utilizador não deve usar muitas letras maiúsculas (dá sensação gritante), usar fontes de letra diferentes ao longo de todo o ficheiro e em caso de tabelas as linhas devem ter cores alternadas.

## Mensagens de diálogo

Quando uma aplicação falha na sua continuidade de estados, causando uma situação de **erro**, esta, pelas heurísticas de usabilidade, deverá ser sempre coberta por **mensagens de erro**. Tais mensagens, apresentadas sob a forma de caixas de diálogo, deverão ter um nível de detalhe do problema algo razoável, no entanto, também não deve conter linguagens descritivas do problema que não façam sentido para o utilizador comum, não-programador. Para tal as mensagens para além de ter de ser curtas devem ser o mais concisas possível.

**erro**  
**mensagem de erro**

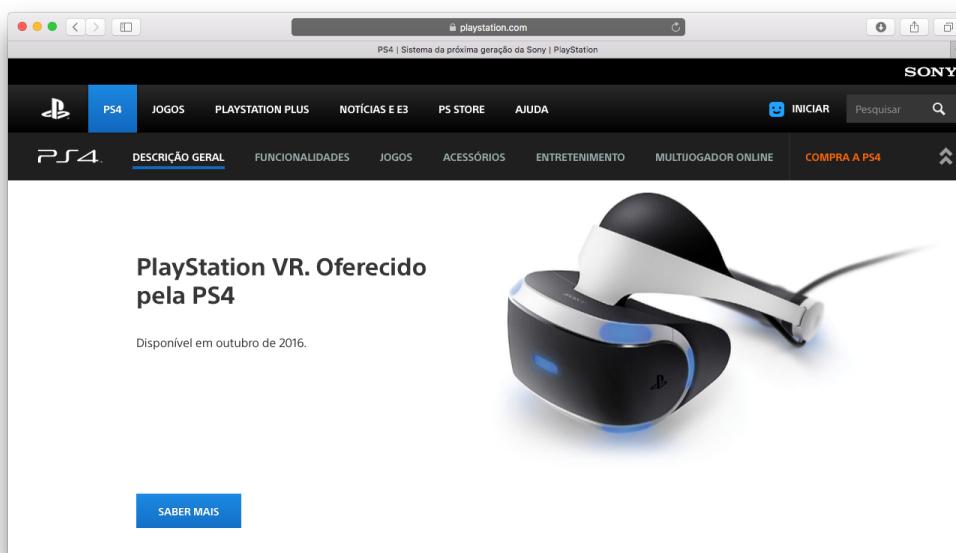
## Paradigma móvel

Nos **dispositivos móveis** a ideia de estrutura deve mudar. Estes dispositivos caracterizam-se, principalmente, pelo tamanho reduzido dos seus ecrãs. No entanto, as páginas web, por exemplo, não podem ser apresentadas, por defeito, com um tamanho à escala do real, dado que depois a página fica demasiado pequena para se conseguir ler o seu conteúdo.

**dispositivos móveis**

A maior parte das *guidelines* são semelhantes tanto para computadores de secretaria, como para dispositivos móveis, no entanto, as interpretações para o paradigma móvel são mais cuidadas: referem-se tópicos mais cuidados a respeito de contexto de utilização, em termos do tamanho do ecrã e de algumas limitações das réplicas das aplicações em plataforma móvel.

Uma das características mais vulgares nos projetos de coexistência em *mobile* e *desktop*, em termos web, por exemplo, têm sido os conteúdos **responsive**, isto é, que se ajustam consoante o tamanho do ecrã. Veja-se a Figura 5.2 e Figura 5.3 onde exibimos a página oficial da PlayStation 4 que tem um comportamento *responsive*.



**figura 5.3**



figura 5.4

## Utilização de cor

A **cor** é um dos componentes que absorvemos com a nossa visão e que mais comportamentos despoleta nos utilizadores. Este, assim sendo, deve ser visto como um assunto de estudo (e em estudo) em várias áreas da ciência como a psicologia, a fisiologia, a física, ou até mesmo na arte, como em *design* gráfico.

Em termos de física dizemos que uma cor é um conjunto de fenómenos de absorção do espetro da luz visível, isto é, reflete a cor que não consegue emitir. Parece antagónico, o pensamento, mas de facto não o é. Assim, uma cor é uma componente que depende das características físicas de um determinado material, da iluminação que a foca, da cor ambiente e do sistema receptor que a identifica - no nosso caso, os nossos **olhos**.

Em termos aplicacionais, para além de aumentar as sensações de realismo, as cores estipulam códigos, alguns deles que involuntariamente os deciframos. Através de uma gradiente de cores podemos identificar itens com mais ou menos importância, atribuindo-lhes ordem, sendo que cada cor possui uma determinada informação a si associada. Muitas vezes é esta cor que, estando predominante num determinado sistema, serve para chamar a atenção aos seus utilizadores ainda que, e mais uma vez, de forma involuntária. Consideremos o seguinte cenário em que temos um reclame à nossa frente, fortemente baseado nas cores vermelho e amarelo - o vermelho ocupa mais espaço que o amarelo. Sobre o que é que é o reclame? Não querendo criar nenhum fenómeno de bruxaria neste documento, mas a maior probabilidade em termos de respostas a esta pergunta será “um reclame de um restaurante”. Vejamos: tentemos lembrar-nos da zona de restauração da grande parte dos centros comerciais ou até mesmo em alguns pontos isolados das cidades. Qual é a cor predominante? Vermelho? Amarelo? Porquê? E porque é que quando pensamos nestas cores nos dá a sensação de **calor**, tal como quando pensamos em tons de verde e azul nos dá a sensação de **frio**? Ora aí está, as cores transportam com elas diversos significados, muitas vezes trazidos em efeito por outras sensações, num processo muitas vezes ligado com fenómenos de **sinestesia** (que apela às emoções).

**cor****olhos****calor**  
**frio****sinestesia**

Numa aplicação comum não devemos então exagerar na **paleta de cores** a usar, salvo casos mais específicos. As cores devem ser usadas de forma muito sábia e sem exageros, pois há muito a perder com o não-estudo dos efeitos das cores em aplicações. No entanto, não podemos dizer que num só produto não se possam utilizar todas as cores que existem - apenas queremos afirmar que as suas escolhas e consequente utilização devem ser coerentes ao longo de todo um projeto.

Mais, as cores podem ter sentidos diferentes de pessoa para pessoa, ainda mais se, sem sabermos, estivermos a lidar com pessoas com distúrbios visuais como o **daltonismo**.

Na página oficial dos Apontamentos (onde estes documentos foram inicialmente publicados) podemos ver uma paleta de 15 cores principais. Nestas cores cada uma identifica, de modo coerente e conciso cada área de estudo do curso de Engenharia de Computadores e Telemática, por conseguinte, de grande parte do mundo da Ciência da Computação. As cores não foram escolhidas ao acaso, tal como a que acompanha a disciplina de Interação Humano-Computador (a3s2) - o rosa. Todas estas cores foram decididas por vias de um longo estudo cromático que permitiu avaliar os impactos das cores pelo nível de atenção exigido para cada área disciplinar. Desde a cor que mais impacto teria na leitura (mais exigente e precisa na interpretação dos textos) - o azul - à que mais impacto teria nas qualidades do desenho e estrutura dos documentos e páginas - o rosa -, todas as cores têm uma justificação cuidada.

## 6. Desenho de Interfaces centrados no Utilizador

Toda esta disciplina anda à volta da importância que um utilizador tem para um sistema interativo. Mas como, em termos gerais, é que podemos desenhar uma aplicação interativa - quais são os processos?

### Desenho de sistemas interativos baseados no utilizador

Até agora vimos que um projeto deve iniciar com um estudo das heurísticas de usabilidade, nos quais abordamos e tentamos aplicar os melhores princípios de usabilidade, paradigmas de usabilidade e outros *standards*. Agora chega a fase em que nos devemos preocupar com a **implementação de um projeto**.

Antes de arrancarmos para uma primeira implementação, devemos re-identificar os casos de sucesso, como é que funcionam e quais são as melhores metodologias de trabalho, para que não hajam razões, mais tarde, para o projeto se afundar.

Na Figura 6.1 podemos ver um esquema de uma metodologia para a implementação de um projeto com base no utilizador. Neste esquema começamos então por definir as características do projeto até atingir os objetivos de usabilidade. Atingidos os objetivos, podemos então seguir para a primeira fase (fase 1) de processamento do projeto. Aqui iremos inicialmente re-engenhar todo o projeto e tentar limar algumas arestas que ainda estejam por limar, de forma a que se consigam extender os objetivos finais do produto final. Seguimos para um **modelo conceptual**, onde já devemos ter delineadas as tarefas vitais da nossa aplicação, de forma a que possamos já passar para o desenho real da aplicação, gerando um **mock-up**. No final desta primeira fase entramos num processo iterativo de **avaliação**, onde tomamos as rédeas do nosso projeto e pomos à prova a estrutura base do projeto já criado (o modelo conceptual).

Numa segunda fase (fase 2), tendo o projeto passado na avaliação da fase 1, entramos agora num processo em que inicialmente criamos *standards* de estruturas e apresentação de itens no ecrã, seguido da geração dos seus protótipos. Tendo realizado tais tarefas, voltamos então ao processo de avaliação, agora da fase 2, que caso falhe volta para o início desta fase.

Numa terceira e última fase (fase 3) o projeto deve receber a implementação mais detalhada em termos de interface e testá-la de seguida. No final, se os objetivos traçados inicialmente não foram alcançados, então deve-se recuar até então e repetir o esquema todo de novo, caso contrário, procede-se para uma fase de testes com utilizadores.

**paleta de cores**

**daltonismo**

**implementação de um projeto**

**modelo conceptual**

**mock-up**

**avaliação**

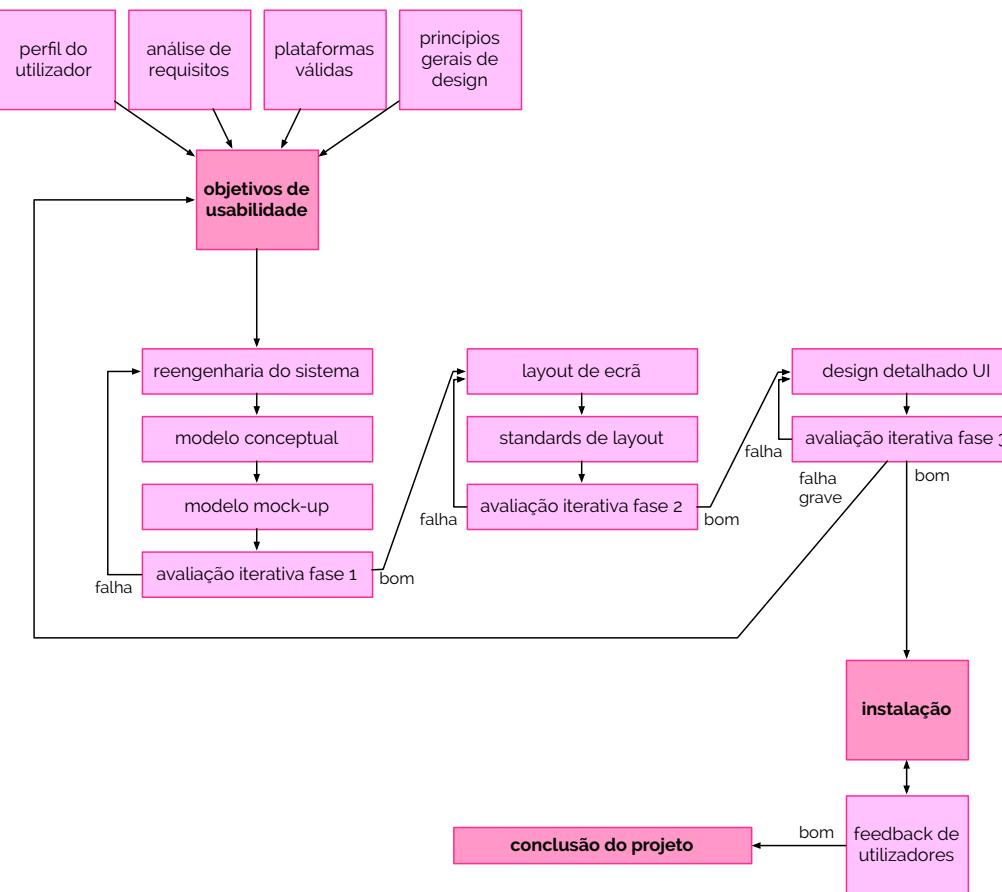


figura 6.1

## Modelos para desenho aplicacional

Para criar uma aplicação interativa necessitamos de uma forma de poder julgar o que está bem e o que não está. Para fazer isso, e sendo que o cliente final deste tipo de aplicações é o utilizador, então precisamos de ter uma forma de obter vários **modelos** para o desenho destas. Tais modelos podem ser mentais, percepionais ou motores.

Em termos de **modelos cognitivos**, estes representam o utilizador na sua interação com o sistema, isto é, modela os aspetos do conhecimento do utilizador e as suas intenções de processamento. O nível de representação varia de modelo para modelo. Um dos exemplos de modelação é o **GOMS** (acrónimo inglês para *Goals, Operators, Methods and Selections*). Na decomposição de uma análise modelar do GOMS temos que este parametriza **objetivos** como o que o utilizador pretende obter, **operadores** como as operações básicas que o utilizador necessita de fazer para usar o sistema, **métodos** como possíveis decomposições de objetivos em sub-objetivos e **selecções** como regras para selecionar os possíveis métodos. Consideremos assim um exemplo: queremos guardar um ficheiro. Para executar tal ação devemos selecionar um método construtor da ação guardar um ficheiro que ativa um determinado botão “guardar”, que abre uma janela, aciona a função que lhe permite carregar em “OK” ou, ao invés de carregar em “guardar”, carrega na tecla de função “ctrl” + “s”. O GOMS também define entre as duas regras criadas qual escolher consoante o estado do utilizador.

Este tipo de modelação permite resolver, em parte, o problema do **fecho**, sobre o qual embora as tarefas ainda não estejam terminadas, se o utilizador a meio de um processo concluir os seus objetivos, sai. Consideremos o caso de um multibanco, em que uma pessoa levanta 20€. Antigamente o cartão só sairia depois do dinheiro. O que acontecia muitas vezes é que as pessoas, ao pegar no dinheiro (objetivo concluído) iam-se embora sem pegar no cartão.

**modelos**

**modelos cognitivos**

**GOMS**

**objetivos, operadores  
métodos  
selecções**

**fecho**

Este procedimento então fica só com uma limitação bem explícita: não há controlo sobre o quanto tempo é que um determinado utilizador demora a executar uma ou mais tarefas.

Um outro modelo é o **KLM** (sigla para *Keystroke-Level Model*) que prevê o desempenho de um utilizador com base nas características **motoras**. O que acontece neste modelo é que são criadas unidades de teste com tarefas interativas simples (cuja duração é inferior a 20s). Estas tarefas têm duas fases: uma fase de **aquisição**, onde o utilizador tem de conseguir aferir uma representação da tarefa, e uma fase de **execução**, onde o utilizador tem de conseguir usar o sistema. No KLM, em si, só é modelada a fase de execução, no entanto esta fase é decomposta em 7 operadores: um **keystroke** (K) (que varia com as habilidades de escrita em teclado), um premir do botão do rato (B), o apontar a um alvo (P), a transição entre o uso do teclado e do rato (H), o desenho do caminho pelo movimento do rato (D), a preparação mental para a ação a ser desenvolvida (M) e a resposta do sistema (R) (que até pode ser ignorada).

Note-se que um dos operadores do KLM era um “apontar ao alvo”. Este operador é uma característica que facilmente modela a capacidade de enquadramento de uma aplicação a uma determinada plataforma, através de uma lei denominada de **Lei de Fitts**. Esta lei estima o tempo médio que um determinado utilizador demora a selecionar um alvo com largura  $W$ , dada uma distância  $D$  deste ao cursor. O cálculo do tempo  $t$  determina-se através de (6.1), onde  $a$  e  $b$  são dois coeficientes.

$$t = a + b \operatorname{lb} \left( 2 \frac{D}{W} \right) \quad (6.1)$$

Um exemplo de aplicação de KLM está no seguinte: consideremos que através do rato, num processador de texto, queremos corrigir um carácter apontando para ele, apagando-o, escrevendo um novo carácter e voltando ao local de partida. Qual o tempo total estimado para esta ação? Vejamos então quantas vezes executamos cada operador. Primeiro temos de mover a mão para o rato (o que perfaz 1 H). De seguida precisamos de colocar o cursor depois do erro (fazendo 1 movimento P e um clique no rato B). Continuando, iremos agora precisar de mover a mão para o teclado (mais 1 H, o que perfaz 2 H) para apagar o carácter e inserir um novo (1 M para pensar no que fazer, 1 K para carregar numa tecla - apagar - e mais 1 K para inserir o novo carácter, perfazendo 1 M e 2 K). Finalmente, recuperar a posição inicial do cursor custar-nos-á mais 1 H (do passar a mão do teclado para o rato) - perfazendo 3 H -, 1 M, para pensar no que fazer a seguir - perfazendo 2 M -, um P para apontar para o sítio onde o cursor estava no início - perfazendo 2 P - e terminando com mais 1 B, por carregar no botão do rato - perfazendo 2 B. Temos assim, no fim, uma estimativa de  $t_{total} = 2(t_K + t_P + t_B + t_M) + 3t_H$ .

Em termos de **análise de tarefas** devemos conseguir, no fim de um processo destes, aferir, entre vários modelos, qual o melhor a seguir. Basicamente, nesta fase tenta-se avaliar como é que as pessoas fazem o seu trabalho - como resolvem tarefas - e o que precisam para as fazer - o que precisam, o que usam e que conhecimento é que precisam de ter antes. Por exemplo, para um utilizador poder ir ao supermercado comprar ovos este desenha que há que sair de casa, entrar no carro, ir até ao supermercado, sair do carro, fechar o carro, entrar no supermercado, comprar os ovos e fazer o contrário para voltar para casa. O utilizador faz então uma **decomposição** de tarefas em componentes unitárias, com base no seu conhecimento - cria uma **hierarquia de tarefas**. A este processo de hierarquização de tarefas damos o nome de **HTA** (sigla inglesa para *Hierarchical Task Analysis*) e serve para criar uma hierarquia de tarefas e sub-tarefas, num planeamento com condições de execução sequencial.

Para decompor uma tarefa em várias até componentes unitárias devemos saber quando parar de o fazer. Assim, cria-se a condição de paragem que, quando a probabilidade de cometer um erro dado um custo for menor que um determinado valor a

**KLM**  
**motoras**

**aquisição**  
**execução**

**keystroke**

**Lei de Fitts**

**análise de tarefas**

**decomposição**  
**hierarquia de tarefas**  
**HTA**

decomposição deve chegar ao seu fim, sendo que esta não deverá ser feita para tarefas cujo nível de complexidade não seja muito baixo, como “esperar 1 minuto”. A avaliação de uma hierarquia de tarefas depende assim muito da experiência de quem o avalia, dado que em algumas tarefas pode haver discrepâncias sobre a forma de resolver problemas de pessoa em pessoa. Note-se que a qualidade dos resultados da análise de tarefas não pode ser melhor que os dados iniciais, dado que esta análise levanta novas questões ao longo do processo, que pode levar a novas resoluções ou criação de novos problemas.

Numa última fase de procedimentos para a implementação de um projeto há que considerar modelos para a avaliação das características de diálogo, fomentando **notações de diálogo**, isto é, descrevendo se a forma como o sistema se apresenta e como ele se comporta é a mais ajustada para toda a sua análise anterior. Aqui geralmente, por vias de estudos semânticos e sintáticos, é possível verificar se um determinado sistema se encontra fechado, isto é, se todas as situações de validade e de erro estão devidamente cobertas por diálogos humano-computador.

**notações de diálogo**

## Métodos de avaliação de interfaces

Como vimos no primeiro capítulo existem normas como a ISO 9241-11, datada de 1998, que estabelece requisitos para equipamentos de escritório ergonómicos com terminais de dados visuais, onde um dos seus capítulos de desenvolvimento é responsável por indicar as melhores diretivas para uma orientação em termos de usabilidade, explicando como identificar quais as informações que um dispositivo nestas condições necessita de seguir para avaliar a usabilidade em termos de desempenho e satisfação. Mas como é que medimos a eficiência e a eficácia de um sistema?

A medição de eficácia e eficiência dependem de análises tanto analíticas (sem utilizadores-teste), como empíricas (com utilizadores-teste). Em ambos os casos considera-se como tarefa impossível uma só pessoa identificar todos os problemas associados com o uso de uma determinada aplicação. Assim, geralmente os projetos tendem a envolver mais pessoas para que seja atingida a eficácia de modo mais significativo. Estas pessoas, em grupo (no máximo aconselhado de 5 pessoas), começam por fazer as **avaliações heurísticas**, fazendo uma primeira análise global, para conhecer a interface, seguindo para uma análise sistemática tendo em conta as heurísticas usadas no projeto, atribuindo graus de severidade a cada fase deste processo. Acontece que usar apenas este processo de avaliação pode tornar toda uma validação do projeto muito subjetiva.

**avaliações heurísticas**

Entram assim os **utilizadores-teste** que devem ser envolvidos no projeto de desenho de interfaces com o seu consentimento, confidencialidade, segurança, liberdade e sem razão para *stress* - é o sistema que está a ser avaliado, não o utilizador. Durante uma designação de tarefas orientadas para o utilizador, este deverá estar a ser avaliado pelos seus comportamentos, de modo direto (em que o observador tira notas), indireto (através de áudio/vídeo), através de entrevista (em que é pedido aos utilizadores que expliquem o que fazem), através de relatórios (*log*) (em que se registam todas as ações do utilizador de forma automática), entre outras.

**utilizadores-teste**

A criação de **relatórios formais** para testes de usabilidade é, normalmente (deve ser normalmente), feita seguindo as normas ISO/IEC 25062:2006 denominadas de **CIF** (acrónimo em inglês para *Common Industry Format*). Tal norma especifica o formato para a criação de relatórios numa apresentação sumária de resultados, especificando mais formalmente os detalhes que falham ao longo de um processo de avaliação, e que podem ser corrigidos. Estes relatórios provêm de **experiências controladas**, nas quais se define uma hipótese, uma entrada, saída e variáveis secundárias, um *design* experimental e se selecionam participantes. Como documentação a preparar há que listar as tarefas a realizar e a respetiva dificuldade, criar um questionário para aferir a avaliação observacional feita ao longo dos testes e listar as tarefas para o observador tirar as respetivas notas.

**relatórios formais**  
**CIF**

**experiências controladas**

## 7. Dispositivos de Entrada e de Saída

Até agora definimos os parâmetros gerais para a criação de um projeto orientado ao utilizador. Mais próximo deste, definimos a interação humano-computador via ecrã, mas este não é o único dispositivo de contacto entre o utilizador e a máquina. Neste capítulo tentamos abordar grande parte de outros dispositivos.

### Dispositivos de entrada

Aos dispositivos que permitem a interação humano-computador no sentido humano para computador damos o nome de **dispositivos de entrada**. Estes podem ser teclados, dispositivos apontadores como ratos, ecrãs táteis, mesas digitalizadoras, câmaras, entre outros.

Comecemos por falar dos **teclados**. Estes dispositivos de entrada são, de todos, os mais utilizados, existindo sob as mais variadas formas e formatos. As grandes variações destes equipamentos revelam-se na posição das teclas ao longo do seu painel. Podendo ter vários tamanhos e aspetos, este é um dos dispositivos que mais tem de variar entre **culturas**, dado que nem todos os países usufruem de um mesmo **alfabeto** ou idioma. Veja-se assim a diferença entre um teclado Português (Portugal) e Inglês (Estados Unidos) na Figura 7.1.



**dispositivos de entrada**

**teclados**

**culturas, alfabeto**

**figura 7.1**

Como podemos reparar na Figura 7.1 grande parte de ambos teclados é igual. Isto acontece porque ambos seguem uma norma denominada de **QWERTY**. A denominação QWERTY provém da ordenação das teclas principais e esta orientação foi criada, embora ainda não se saiba com certezas, porque a proximidade de teclas cujas letras frequentemente se usam juntas fazia com que as varetas das máquinas de escrever (com teclados antigos) bloqueassem, por vezes partindo as varetas, para quem escrevia mais depressa. Alguns países não usam o *layout* QWERTY, mas antes **AZERTY**, como em França, como podemos ver na Figura 7.2. Outros *layouts* incluem **Dvorak** cujo desenho reduz a necessidade de movimentar as mãos para escrever as palavras, que podemos ver na Figura 7.3.

**QWERTY**

**AZERTY**

**Dvorak**



**figura 7.2**



**figura 7.3**

Um outro dispositivo bastante usado é o **rato**. Inventado por Engelbart, este dispositivo funciona de forma mecânica e transporta indiretamente um movimento para **apontar** um cursor que se apresenta no ecrã. Na Figura 7.4 podemos ver o rato de Engelbart.

**figura 7.4**



Este dispositivo aponta muitas qualidades, mas o seu uso contínuo, tal como o do teclado, pode gerar problemas motores ao utilizador nas mãos, dado que necessita de grandes movimentos das mãos e espaço na mesa. Uma outra limitação, embora nos dias que corram já não se coloque, é a coordenação entre os olhos e o gesto da mão com o rato, para apontar indiretamente para objetos no ecrã.

Outros dispositivos de entrada passam por *trackballs*, *joysticks* e mesas digitalizadoras (e ecrãs táteis), e reconhecimento por voz, aspeto já referido anteriormente.

## Dispositivos de saída

Contrariamente aos dispositivos de entrada, os **dispositivos de saída** são **dispositivos de saída** equipamentos que potenciam o diálogo entre o computador e o humano. Alguns exemplos destes são os ecrãs, como já referimos anteriormente numa unidade anterior, e auscultadores.

E assim terminamos a disciplina de Interação Humano-Computador (a3s2) dando agora continuidade na aplicação destes conteúdos na disciplina de Computação Visual (a4s1).



**1. Princípios e Paradigmas de Usabilidade**

Normas de utilização.....	2
História em termos de paradigmas de usabilidade .....	3

**2. Introdução às Interfaces de Utilizador**

Princípios de usabilidade.....	4
--------------------------------	---

**3. O Utilizador e Modelos Mentais**

Perfil de utilizador.....	7
Influências do utilizador no desenho de interfaces.....	10
Modelos mentais de utilizador.....	10

**4. Diálogo entre computador e humano**

Classe de diálogo .....	12
Menus.....	12
Linguagens naturais.....	16
Linguagem de comandos .....	18
Formulários.....	19
Teclas de função .....	20
Manipulação direta.....	20

**5. Apresentação em Ecrã**

Estrutura (Layout) .....	21
Mensagens de diálogo.....	22
Paradigma móvel.....	22
Utilização de cor.....	23

**6. Desenho de Interfaces centrados no Utilizador**

Desenho de sistemas interativos baseados no utilizador.....	24
Modelos para desenho aplicacional.....	25
Métodos de avaliação de interfaces.....	27

**7. Dispositivos de Entrada e de Saída**

Dispositivos de entrada .....	28
Dispositivos de saída .....	29



## **Apontamentos de Interação Humano-Computador**

1<sup>a</sup> edição - junho de 2016

ihc

**Autor:** Rui Lopes

**Fontes bibliográficas:** "Human Computer Interaction", DIX, A., FINLEY, J., RUSSELL, B., 3a edição; "Designing the User Interface - Strategies for Effective Human-Computer Interaction", SHNEIDERMAN, B., PLAISANT, C., COHEN, M., JACOBS, 5a edição.

**Outros recursos:** Aulas de Interação Humano-Computador

**Agradecimentos:** Professor Joaquim Madeira e Beatriz Sousa Santos.

Todas as ilustrações gráficas são obra de Rui Lopes e as imagens são provenientes das fontes bibliográficas divulgadas.



**apontamentos**

© Rui Lopes 2016 Copyright: Pela Creative Commons, não é permitida a cópia e a venda deste documento. Qualquer fraude será punida. Respeite os autores e as suas marcas. Original - This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit [http://creativecommons.org/licenses/by-nc-nd/4.0/deed.en\\_US](http://creativecommons.org/licenses/by-nc-nd/4.0/deed.en_US).